

## ĐỀ THI HỌC KỲ HỆ ĐIỀU HÀNH

**Câu 1.** Trong hệ thống quản lý bộ nhớ ảo chỉ dùng phân trang với kích thước trang là 1024 bytes. Cho bảng phân trang process như sau

Page	Frame
0	4
1	2
2	6
3	1

Chỉ số trang ảo được đánh số từ 0 ...3. Các giá trị tính theo hệ thập phân. Hãy chọn phát biểu đúng

- A. Tương ứng với địa chỉ luyện lý 3022 là địa chỉ vật lý 7118
- B. Tương ứng với địa chỉ luyện lý 1901 là địa chỉ vật lý 877
- C. Tương ứng với địa chỉ luyện lý 100 là địa chỉ vật lý 4100
- D. Tương ứng với địa chỉ luyện lý 4096 là địa chỉ vật lý 2048

**Câu 2.** Cho chuỗi tham chiếu nhớ như sau

Time	0	1	2	3	4	5	6	7	8	9	10	11
Page	1	2	3	4	4	2	5	3	2	5	6	4

Sử dụng giải thuật thay trang LRU, mỗi quá trình được cấp phát 5 frame trống. Các trang trong bộ nhớ chính sau khi thực hiện tham chiếu trang nhớ ở thời điểm 10 (trang 6) là:

- A. 6-2-3
- B. 2-6-5
- C. 6-5-2
- D. 5-6-2

**Câu 3.** Giải thuật định thời CPU nào dưới đây phù hợp cho các ứng dụng IO-bound?

- A. RR với quantum time ngắn
- B. SJF
- C. FCFS
- D. SRTF

**Câu 4.** Cơ chế ánh xạ thread người dùng (user thread) vào thread lõi (kernel thread) nào sau đây tổng quản lý luồng thread để không bị trình trạng “một thread bị block thì tất cả thread khác cũng bị ảnh hưởng”?

- A. One to one và many to many
- B. Many to one và many to many
- C. Các cơ chế trong câu hỏi khác đều được
- D. Many to one và one to one

**Câu 5.** Việc gán địa chỉ (address binding) được thực hiện bởi trình biên dịch là quá trình

- A. Chuyển đổi địa chỉ khả tái định vị (relocatable address) thành địa chỉ tuyệt đối

- B. Chuyển đổi địa chỉ dạng ký hiệu (symbollic address) thành địa chỉ tuyệt đối.
- C. Chuyển đổi địa chỉ dạng ký hiệu (symbollic address) thành địa chỉ khả tái định vị (relocatable address)
- D. Gán một tập tin thực thi vào một không gian bộ nhớ tuyệt đối

**Câu 6.** Các dịch vụ của hệ điều hành nhằm cung cấp các chức năng nào sau đây để hướng quản lý hệ thống một cách hiệu quả

- A. Giao tiếp quá trình, giao diện làm việc với người sử dụng, và sử lý hệ thống tập tin
- B. Cấp phát tài nguyên, quản lý tài khoản và bảo mật bảo vệ
- C. Các lựa chọn khác điều sau
- D. Thực thi chương trình, xử lý hệ thống tập tin, phát hiện lỗi

**Câu 7.** Cho chuỗi tham chiếu nhớ như sau

Time	0	1	2	3	4	5	6	7	8	9	10	11
Page	1	2	3	1	4	2	5	3	2	5	6	4

Sử dụng giải thuật thay trang FIFO, mỗi quá trình được cấp phát 3 frame trống. Các trang trong bộ nhớ chính sau khi thực hiện tham chiếu trang nhớ ở thời điểm 8 (trang 2) là:

- A. 4-5-2
- B. 5-2-3
- C. 5-3-2
- D. 5-4-2

**Câu 8.** Với cơ chế standard swapping, để kích hoạt việc swap một ảnh của quá trình ra khỏi bộ nhớ thì hệ điều hành phải dựa trên sự chuyển trạng thái nào của quá trình

- A. Ready->running
- B. Running->ready
- C. Running->terminated
- D. Running->waiting

**Câu 9.** Cho một hệ thống gồm 4 process có thông tin như trong bảng dưới đây. Các process được cấp phát CPU theo giải thuật định thời Shortest Remaining Time First (SJRT), process đến trước sẽ được ưu tiên. Tại thời điểm 4 quá trình nào đang thực thi

Process	Thời điểm đến	Burst time
P1	0	6
P2	1	3
P3	2	1
P4	3	2

Chọn một

- A. P3
- B. P1
- C. P2
- D. P4

**Câu 10.**

```
Process p0
{
    While (true)
    {
        Wait(s0);
        Print"Hello"
        Signal(s1);
        Signal(s2);
    }
}
Process p1
{
    Wait(s1);
    Signal(s0);
}
Process p0
{
    While (true)
    {
        Wait(s2);
        Signal(s0);
    }
}
```



“hello” sẽ được in ra bao nhiêu lần

- A. Ít nhất 2
- B. Ít nhất 3**
- C. 2 lần
- D. 3 lần

**Câu 11.** Tại sao các thread lại cần phải có thanh ghi (register) và stack riêng (không giống như phần code, data, files) khi quản lý

- A. Giúp cho hệ điều hành ghi nhớ được thread nào đang được gán trên bộ xử lý hoặc lõi nào
- B. Các thread thực thi trên các bộ xử lý hoặc lõi khác nhau nên cần lưu riêng
- C. Các thread khác nhau thực thi độc lập và đây là hai thành phần chính kiểm soát thực thi**
- D. Lưu thanh ghi và stack riêng giúp thread tăng tốc được quá trình xử lý

**Câu 12.** Khi nào thì xảy ra hiện tượng phân mảnh trên hệ thống tập tin

- A. Không gian chưa sử dụng trên đĩa làm việc
- B. Không gian đã sử dụng trên đĩa không liên tục
- C. Các tập tin không liên tục nhau
- D. Không gian chưa sử dụng trên đĩa không liên tục**

**Câu 13.** Tổ chức tập tin và thư mục trên một phân hoạch đĩa được lưu trữ trong

- A. Cấu trúc thư mục
- B. Khối điều khiển tập tin FCB
- C. Bảng mount
- D. Khối điều khiển volume

**Câu 14.** Cho chương trình có đoạn mã như sau

```
/* shared variable */
boolean  choosing[ n ]; /* initially, choosing[ i ] = false */
int      num[ n ]; /* initially, num[ i ] = 0 */
do {

    num[ i ] = max(num[0], num[1], ..., num[n - 1]) + 1;

    for (j = 0; j < n; j++) {
        while (choosing[ j ]);
        while ((num[ j ] != 0) && (num[ j ] < (num[ i ], 0)));
    }
    /* CRITICAL SECTION */
    num[ i ] = 0;
    remainder section
} while (1);
```

**TÀI LIỆU SƯU TẬP**  
BỞI HCMUT-CNCP

- A. Có thể có đúng 2 quá trình được vào cùng tranh chấp tại một thời điểm
- B. Chỉ có 1 quá trình được thực thi vùng tranh chấp tại một thời điểm
- C. Không thỏa mãn tính chất tiến triển và tính chất không có trì hoãn vô hạn định
- D. Có thể có nhiều hơn 2 quá trình được vào vùng tranh chấp tại một thời điểm

**Câu 15.** Chọn phát biểu đúng về cấu trúc monolithic

- A. Việc thêm các chức năng của hệ điều hành được thực thi dễ dàng hơn so với cấu trúc microkernel
- B. Các dịch vụ của hệ điều hành đều được tích hợp vào kernel
- C. Có hiệu năng giảm so với các cấu trúc khác
- D. Việc trao đổi dữ liệu giữa các thành phần chức năng của kernel thực hiện chủ yếu bằng cơ chế message-passing

**Câu 16.** Chờ đợi có giới hạn (bounded waiting) nghĩa là tồn tại một giới hạn về số lần một quá trình được cho phép vào vùng tranh chấp

- A. Sau khi một quá trình bất kì yêu cầu vào vùng tranh chấp và trước khi yêu cầu được chấp nhận
- B. Sau khi các quá trình khác đều hoàn tất n lần vùng tranh chấp
- C. Trước khi một quá trình yêu cầu vào vùng tranh chấp
- D. Khi một quá trình khác đang trong vùng tranh chấp

**Câu 17.** Tập tin nào là gồm một chuỗi các byte được tổ chức theo khối và được hiểu bởi linker của hệ thống

- A. Tập tin object
- B. Tập tin văn bản
- C. Tập tin mã nguồn
- D. Tập tin thực thi được

**Câu 18.** Điểm nào sau đây là lợi ích của việc dùng 2 bảng mở tập tin (open file table) trong lõi hệ điều hành khi hiện thực một hệ thống tập tin

- A. Các câu khác đều sai
- B. Để hỗ trợ truy xuất hiệu quả trong trường hợp có nhiều tác vụ mở cùng một tập tin
- C. Để triển khai cơ chế điều khiển quyền truy xuất
- D. Để di chuyển bảng mở tập tin cho từng tập tin (per-process open file table) vào trong không gian của người dùng nhằm tiết kiệm bộ nhớ trong lõi

**Câu 19.** Hệ thống tập tin nào sau đây không phải thuộc dạng địa phương (local – chỉ quản lý đĩa cứng cục bộ)

- A. FAT
- B. NTFS
- C. UFS
- D. NFS

**TÀI LIỆU SƯU TẬP**  
BỞI HCMUT-CNCP

**Câu 20.** Một tập các quá trình sử dụng semaphore để đồng bộ với giá trị khởi tạo là 5. Các quá trình gọi lần lượt 9 lần wait(), 3 lần signal(). Hãy chọn phát biểu đúng sau khi các quá trình hoàn tất gọi signal() và wait()

- A. Không có quá trình nào đang đợi trên semaphore
- B. Có 5 quá trình có thể đồng thời thực thi vùng tranh chấp tại cùng một thời điểm
- C. Nếu có yêu cầu mới đi vào vùng tranh chấp, có duy nhất 1 yêu cầu được đáp ứng
- D. Có 1 quá trình đang đợi trên semaphore

**Câu 21.** Cho biết tỉ lệ lỗi trang là 0,3 – thời gian truy xuất bộ nhớ là 20ns, thời gian swap in out là 100ns, thời gian xử lý lỗi trang là 5ns. Tính thời gian truy xuất hiệu dụng

- A. 75,5
- B. 45,5  $0,3*(100+5) + 0,7*20$
- C. 71,5
- D. 149,5

**Câu 22.** Ngắt gì được sử dụng để phục vụ quá trình định thời hệ điều hành

- A. Ngắt đồng bộ và ngắt IO
- B. Ngắt mềm (trap)
- C. Ngắt đồng hồ (timer)
- D. Ngắt IO

**Câu 23.** Thrashing xảy ra khi

- A. Process được tạo ra và kết thúc liên tục
- B. Địa cứng không còn khoảng trống cho swap file
- C. Tổng lượng bộ nhớ của các process yêu cầu lớn hơn kích thước bộ nhớ vật lý
- D. Tổng lượng bộ nhớ trong các working set lớn hơn kích thước bộ nhớ vật lý

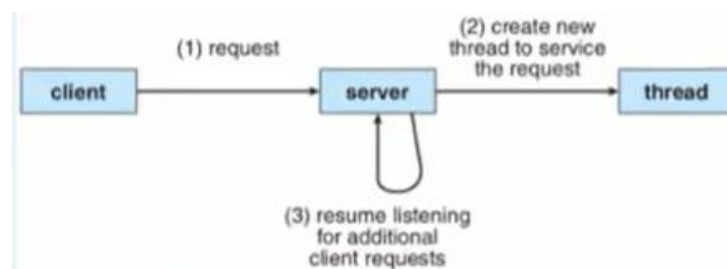
**Câu 24.** Hệ thống làm việc theo nguyên tắc phân chia thời gian (time sharing system) phù hợp với các ứng dụng/ công việc

- A. Thời gian thực (real time job)
- B. Hướng tương tác (interactive job)
- C. Thiên về tính toán CPU
- D. Bó công việc

**Câu 25.** Trong các cơ chế cấp phát sau, cơ chế nào giúp truy cập file ngẫu nhiên nhanh nhất

- A. Cấp phát theo danh sách kề
- B. Cấp phát liên tục
- C. Cấp phát dùng chỉ mục
- D. Cấp phát dùng chỉ mục nhiều mức

**Câu 26.** Apache (một máy chủ web rất phổ biến) sử dụng cơ chế one thread per connection (một thread một kết nối) để phục vụ các yêu cầu đến giống như trong hình sau



Giả thiết số lượng các yêu cầu kết nối đến máy chủ web dùng apache lên đến vài trăm nghìn yêu cầu đồng thời. Hãy cho biết cơ chế one thread per connection có còn hợp lý hay không

- A. Hợp lý. Thời gian chuyển đổi ngữ cảnh giữa các thread là nhỏ nên không ảnh hưởng tới thời gian đáp ứng các yêu cầu
- B. Không hợp lý. Số lượng thread tạo ra quá lớn, giữ nhiều tài nguyên trong khi nhiều kết nối web không thực sự có nhiều tương tác

- C. Không hợp lý. Số lượng thread tạo ra quá lớn, dẫn đến gian đáp ứng với yêu cầu mới sẽ sụt giảm
- D. Hợp lý. Số lượng thread lớn sẽ phân bổ cân bằng dẫn đến các bộ xử lý giúp đáp ứng hiệu quả

**Câu 27.** Nội dung nào sau đây không phải là nhược điểm của dạng cấp phát liên tục khi hiện thực hệ thống tập tin

- A. Chi phí lớn khi dịch chuyển đầu đọc tới lui trong một tập tin
- B. Có thể tồn tại phân mảnh ngoài
- C. Có thể tồn tại phân mảnh trong nếu có hiện thực một cơ chế cấp phát sớm (pre-allocated mechanism)
- D. Những tập tin không thể mở rộng về mặt kích thước

**Câu 28.** Chọn phát biểu đúng

- A. Bất thường Belady là khi giảm lượng frame cấp phát thì số lần page\_fault tăng
- B. Số lần page fault phụ thuộc chủ yếu vào các yếu tố: lượng frame được cấp phát và giải thuật thay trang
- C. Trong một giải thuật thay trang bất kỳ, khi tăng lượng frame cấp phát thì số lần xảy ra page fault luôn luôn giảm
- D. Bất thường Belady có thể xảy ra ở giải thuật FIFO và giải thuật CLOCK

**Câu 29.** Cho một hệ thống gồm 5 process có thông tin như trong bảng dưới đây. Các process được cấp phát CPU theo giải thuật định thời SJF, thời gian ra quyết định trì hoãn là 1 giây. Tính thời gian đợi trung bình của các process

Process	Thời điểm đến	Burst time
P1	0	8
P2	0,1	4
P3	0,5	2
P4	0,8	1

- ☒ A. 3,75
- ☐ B. 3,4
- C. 2,75
- D. 4,4

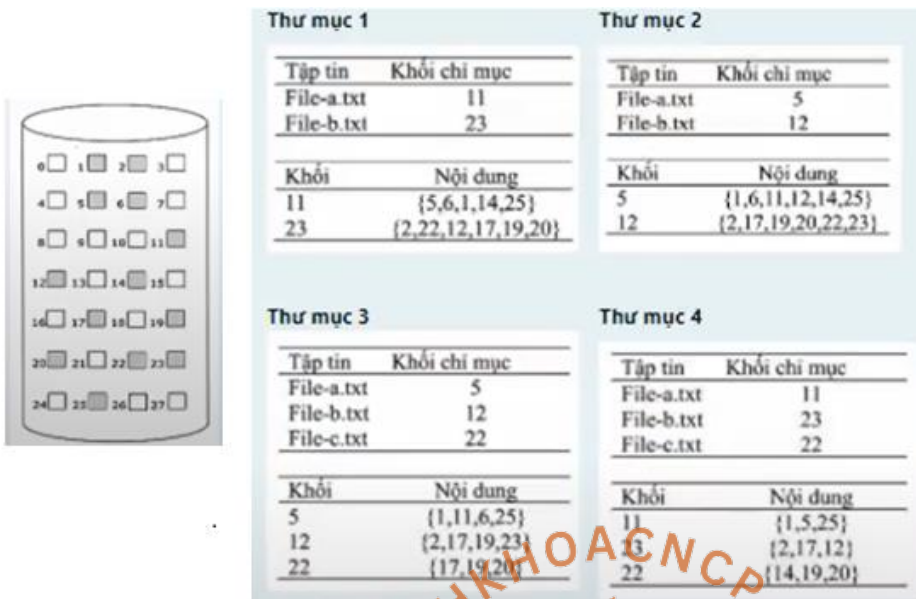
**Câu 30.** Những kỹ thuật nào sau đây có thể được sử dụng để nạp những chương trình lớn hơn khả năng của bộ nhớ, mà không cần sự can thiệp của người lập trình?

- A. Liên kết tĩnh
- B. Liên kết động
- C. Nạp động
- D. Tất cả các đáp án

**Câu 31.** Một hệ thống tập tin được hiện thực theo cơ chế chỉ mục (indexed allocation) như hình vẽ. Những khối máy xám là đã được cấp phát cho các tập tin. Câu trả lời nào sau đây là đúng cho



cấu trúc thư mục (directory structure) và những khối chỉ mục (index block) của hệ thống tập tin đó



- A. 1
- B. 2
- C. 3
- D. 4

**Câu 32.** Trong việc gán địa chỉ (address binding), thanh ghi tái định vị (relocation register) được dùng để ánh xạ giữa địa chỉ logic (logical address) và địa chỉ ảo (virtual address) khi nào?

- A. Lúc nạp chương trình
- B. Lúc biên dịch
- C. Lúc liên kết
- D. Lúc thực thi

**Câu 33.** Định nghĩa nào sau đây đúng với một hệ thống tập tin dạng journaling

- A. Một sự liên kết giữa cấp phát dạng liên kết (linked allocation) và cấp phát dạng chỉ mục (indexed allocation)
- B. Một kỹ thuật để lưu lại (log) tất cả những giao dịch tập tin/ thư mục giúp khả năng phục hồi
- C. Một ý tưởng để cải tiến dạng cấp phát liên tục (continuous allocation) nhằm tránh phân mảnh ngoại
- D. Một phương pháp để theo vết tất cả những khối dữ liệu chưa dùng trên đĩa

**Câu 34.** Cho một hệ thống gồm 4 process có thông tin như trong bảng dưới đây. Các process được cấp phát CPU theo giải thuật định thời Shortest Remaining Time First (SJRT), process đến trước sẽ được ưu tiên. Tính thời gian đợi trung bình



Process	Thời điểm đến	Burst time
P1	0	6
P2	1	3
P3	2	1
P4	3	2

Chọn một

- A. 1,25
- B. 2,75
- C. 3,25
- D. 2,25**

**Câu 35.** Cho chuỗi tham chiếu nhớ như sau

Time	0	1	2	3	4	5	6	7	8	9	10	11
Page	1	2	3	1	4	2	5	3	2	5	6	4

Sử dụng giải thuật thay trang Optimize, mỗi quá trình được cấp phát 3 frame trống. Các trang trong bộ nhớ chính sau khi thực hiện tham chiếu trang nhớ ở thời điểm 9 (trang 5) là:

- A. 5-3-2**
- B. 2-3-5
- C. 4-5-2
- D. 5-2-3**

**Câu 36.** Một tập tin thực thi sở hữu root (ví dụ như ping). Cách nào để lỗi hệ điều hành cho phép một user bình thường có thể thực hiện file thực thi này

- A. Hệ điều hành không cần làm gì hết
- B. Hệ điều hành sẽ thay đổi user ID của quá trình sang Effective ID trùng với ID của tập tin thực thi**
- C. Hệ điều hành thay đổi user ID của tập tin sang user của người dùng bình thường, rồi thực thi tập tin này
- D. Chỉ có root mới thực thi được tập tin đã cho

**Câu 37.** Cho một hệ thống 5 process. Tất cả đều vào hệ thống tại thời điểm 0. Tính thời gian đợi trung bình khi dùng giải thuật RR, quantum time là 1

Process	Burst time
P1	5
P2	3
P3	6
P4	2
P5	1

- A. 9
- B. 6
- C. 7
- D. 8**

**Câu 38.** Để định thời hệ thống process về tính toán (CPU – intensive process), giải thuật định thời CPU nào hợp lý nhất

- A. SJF
- B. SRTF
- C. P (non-preemptive)**
- D. RR

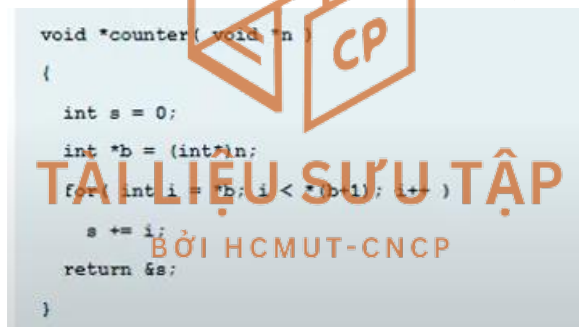
**Câu 39.** Chọn đúng về chuyển ngữ cảnh đối với user-level thread

- A. Chuyển ngữ cảnh giữa hai thread không tốn CPU
- B. Chuyển ngữ cảnh giữa hai thread do bộ định thời thực hiện
- C. Chuyển ngữ cảnh giữa hai quá trình tốn ít chi phí sử dụng CPU hơn giữa hai thread
- D. Chuyển ngữ cảnh giữa hai thread tốn ít chi phí sử dụng CPU hơn giữa hai quá trình**

**Câu 40.** Giao diện (interface) cung cấp để truy cập các dịch vụ của hệ điều hành thông qua

- A. System calls**
- B. Application programming interface
- C. Lệnh mã máy
- D. Thư viện hàm

**Câu 41.** Xem xét một chương trình C như bên dưới



```
void *counter( void *n )
{
    int s = 0;
    int *b = (int*)n;
    for( int i = *b; i < *(b+1); i++ )
        s += i;
    return &s;
}
```

Hãy cho biết hàm counter() có vấn đề gì

- A. Giá trị ở vùng nhớ b+1 không được dùng
- B. Hàm không có vấn đề gì
- C. Trả về một con trỏ đến tổng s là không cần thiết, mặc dù vẫn dùng được
- D. Trả về con trỏ biến địa phương (s) là không ổn định**

**Câu 42.** Chọn câu đúng về so sánh multiprocessing và time-sharing (multi tasking)

- A. Multiprogramming làm giảm hiệu suất sử dụng CPU còn time-sharing thì làm tăng hiệu suất sử dụng
- B. Hệ thống Multiprogramming không chú trọng khả năng tương tác hiệu quả với người dùng, còn time-sharing không cung cấp khả năng tương tác hiệu quả với người dùng.**
- C. Trái ý B
- D. Multiprogramming giữ nhiều công việc trong bộ nhớ còn time-sharing chỉ giữ một

**Câu 43.** Cho bảng sau

Job	Bộ nhớ yêu cầu (KB)	Thời gian bắt đầu (phút)	Thời gian kết thúc (phút)
Job 1	50	0	70
Job 2	30	10	30
Job 3	25	15	33
Job 4	15	45	60
Job 5	30	50	70

Giả sử bộ nhớ máy tính chạy những job trên chỉ có kích thước 100KB và được quản lý theo dạng cấp phát liên tục (continuous memory allocation) với phân hoạch kích thước thay đổi (variable-partition). Bộ nhớ chưa sử dụng sẽ được cấp phát từ thấp đến cao theo cơ chế phù hợp nhất trước (first-fit-strategy). Tại phút 55, trong bộ nhớ còn lại job nào?

- A. Job 1, Job 4
- B. Job 4, Job 5
- C. Job 1, Job 4, Job 5**
- D. Job 1, Job 3, Job 4

**Câu 44.** Chọn đúng

- A. Khoảng thời gian ngắt quãng của bộ đếm thời gian (fixed-interval timer interrupt) càng lớn càng ước lượng chính xác các trạng nhớ trong working set
- B. Kích thước cửa sổ delta trong mô hình working set càng lớn càng giúp phát hiện được tốt tập trạng nhớ tham chiếu cục bộ (locality)**
- C. Mô hình working set cho phép ước lượng động số frame mà quá trình cần theo các thời điểm để hạn chế lỗi trang
- D. Các trang nhớ sẽ được tham chiếu trong tương lai gần được dự đoán bởi mô hình working set

**Câu 45.** Cho trước vùng nhớ [0,999] được dành cho những quá trình người dùng. Có những phần của vùng nhớ này được cấp trước cho các quá trình chạy rất lâu ở các đoạn địa chỉ [200,249], [400, 599], [900,949]. Có một số chương trình mới đến, chi tiết như sau

Quá trình	Bộ nhớ yêu cầu	Thời điểm bắt đầu	Thời điểm kết thúc
1	200	0	10
2	250	5	25
3	150	18	40
4	100	22	35

Bộ nhớ được quản lý theo cơ chế phân hoạch thay đổi (variable-partition). Giả thiết bộ nhớ sẽ đáp ứng yêu cầu mới dựa trên nguyên tắc phù hợp đầu tiên (first-fit). Khi đó, những lỗ hổng trong bộ nhớ tại thời điểm 24 là

- A. [0,199], [350,399], [750,899], [950,999]
- B. [150,199], [350,399], [850,899], [950,999]**

- C. [150,199], [250,399], [850,899], [950,999]
- D. [150,199], [350,399], [850,999]

**Câu 46.** Cho 3 cv A, B, C có độ ưu tiên cao, trung bình, thấp. A và C có thao tác lên biến chia sẻ. Cv A gián tiếp lấy quyền thực thi bởi B vì C đang trong vùng tranh chấp nên A phải đợi. Tình huống này được xử lý bằng cách cho phép A nhường lại độ ưu tiên cho C để C tiếp tục thực thi và kết thúc vùng tranh chấp. Kỹ thuật này gọi là gì?

- A. Đảo ngược độ ưu tiên
- B. Nâng cấp độ ưu tiên
- C. Chỉnh sửa độ ưu tiên
- D. Loại bỏ độ ưu tiên

**Câu 47.** Một thread bao gồm

- A. PC, stack, register và signal
- B. PC, code
- C. PC, code, stack
- D. PC, stack, register

**Câu 48.** Chọn đúng

- A. Truy xuất ngẫu nhiên phù hợp cho cả dạng cấp phát theo chỉ mục và danh sách liên kết
- B. Truy xuất tuần tự có thể thực hiện được trong tất cả chiến lược cấp phát không gian lưu trữ cho tập tin
- C. Truy xuất tuần tự chỉ thực hiện được nếu tập tin được cấp phát theo không gian liên tục
- D. Truy xuất ngẫu nhiên cho thời gian truy xuất tập tin nhanh hơn truy xuất tuần tự trong các chiến lược cấp phát không gian lưu trữ cho phép truy xuất ngẫu nhiên

**Câu 49.** Xét một hệ thống tập tin FAT. Ý tưởng nào sau đây có thể được sử dụng để phục hồi tập tin (những khối không bị hỏng) nếu một vài khối của tập tin đó bị hỏng

- A. Sử dụng doubly linked list
- B. Lưu thêm cấu trúc thư mục để phục hồi
- C. Lưu một danh sách các khối cho tập tin đó trong cấu trúc thư mục
- D. Các câu khác sai

**Câu 50.** Cho trước hình ảnh về cấp phát khối dữ liệu trên đĩa. Cơ chế nào phù hợp nhất đã thực hiện cho hệ thống tập tin

- A. Cấp phát liên kết
- B. Cấp phát chỉ mục
- C. Cấp phát chỉ mục nhiều mức
- D. Cấp phát liên tục

